CSC 310, Fall 2011 — Theory Assignment #1

Due in tutorial on October 7. Worth 7% of the course grade.

Note that this assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own.

Question 1 (15 marks): Consider a source with an alphabet of three symbols, a_1, a_2, a_3, with probabilities p_1, p_2, p_3. Suppose we use a code in which the codewords for a_1, a_2, a_3 are $0, 10, 11$. Describe and plot (by hand is fine) the set of values for p_2 and p_3 for which this is an optimal code. (The value of p_1 is of course determined from these as $p_1 = 1 - p_2 - p_3$.)

Question 2 (20 marks): Consider a source with a six-symbol alphabet, a_1, a_2, a_3, a_4, a_5, a_6, with probabilities $p_1 = 0.2$, $p_2 = 0.01$, $p_3 = 0.35$, $p_4 = 0.38$, $p_5 = 0.02$, $p_6 = 0.04$.

a) Find the entropy of this source.

b) Find a Huffman code for this source.

c) Compute the expected code length of this Huffman code.

Question 3 (20 marks): Suppose we have a source with alphabet $a_1, a_2, \ldots, a_I$, for which the symbols have probabilities $p_1, p_2, \ldots, p_I$. Assume that $p_1 \geq p_2 \geq \cdots \geq p_I$, and that none of the p_i are zero. The coding alphabet is $\{0, 1\}$.

Consider the following conjecture:

> Let C be any uniquely-decodable code for this source in which the code words for $a_1, a_2, \ldots, a_I$ have lengths $l_1, l_2, \ldots, l_I$, and let C' be any other uniquely-decodable code for this source, with codeword lengths of $l'_1, l'_2, \ldots, l'_I$ for $a_1, a_2, \ldots, a_I$. If C and C' are both optimal, then $l_i = l'_i$ for $i = 1, \ldots, I$.

Either prove that this conjecture is true, or demonstrate that it is false by giving an explicit counterexample.

Question 4 (45 marks): Recall the run-length encoding scheme described in the first lecture: We wish to encode a sequence of 500 black or white pixels, in which we expect long runs of black pixels and long runs of white pixels. The first bit we send is 0 if the first run is white, or 1 if the first run is black. We then send a series of 3-bit counts, which are the lengths of runs. If the count is less than 7, the following run is of the opposite colour. For a run with a count of 7 (the maximum specifiable in 3 bits), the next run is the same colour. (Note that a run of exactly 7 pixels would be encoded as a run of length 7 followed by a run of length 0). For example, if the initial pixels are `BBBBBBWWWWBBBBBBBBBWWWB...`, the initial bits sent will be 1 110 100 111 001 011...

We can consider this run-length encoding scheme as a way of encoding a single symbol from an alphabet of 2^{500} symbols (one symbol for every possible image with 500 black or white pixels).

The lecture notes give an example of a message with three counts of 3, 0, and 2 pixels (ie, code bits 011 000 010) which will never be produced, since a run of 5 pixels would instead be coded as a single count of 5 (ie, as code bits 101). This illustrates that the code can't be optimal. It can be improved as explained in the lecture notes by getting rid of all nodes in the code tree that have only one child. (There are in fact *two* ways in which this code is inefficient in this respect.)

a) Describe (in as simple terms as you can) what the coding procedure will be once nodes with only one child are eliminated from the code tree.

b) What is the maximum number of bits by which the modification in part (a) shortens a codeword for some image?

c) For this part, assume that the image is not 500 pixels long, but is instead very, very long, so that how a run near the end of the image is coded is of negligible importance. Suppose that the length of a run of white or black pixels is uniformly distributed from 1 to 10 (ie, each of these values has probability 1/10). If the image has n runs (assumed to be very big), what will be the expected number of bits in the encoding if the original code is used, and if the coded as modified in part (a) is used?

d) Find another simple modification to the original code for which the expected number of bits used to encode n runs, using the uniform distribution of run lengths between 1 and 10 as for part (c)), is smaller than for the modified code you found in part (a).